

REMARKS:

In response to the Examiner's Office Action of September 21, 2006, Applicant is herein presenting their considerations and response to the Examiner's comments.

Referring in particular to the Office Action, the claims have been amended to address the claim rejections under 35 U.S.C. Section 112. In particular, the phrase "may be obtained" has been amended to remove the indefinite nature of the claim. Moreover, the preamble of Claim 3 has been amended to clearly define Claim 3 as a method claim. Applicant submits that the claims are now in order and all formality objections have been overcome.

Referring to the substantive objections taken to the claims, Examiner contends that Claims 1 to 3, 6 to 8 and 10 to 17 are anticipated by *Krychniak* (US 6,192,357).

Applicant takes this opportunity to reiterate the underlying inventive concept of the invention, which is brought out in each of the independent claims. Firstly, the claimed invention defines a method and system which preferably avoids the use of a "join operation" when extracting data from a database. This process is explained generally at Page 12, beginning Line 4 and ending Line 16.

Referring to the abovementioned description, the claimed invention avoids the need for the use of a "join operation" by creating an additional entity (such as a table), which contains aggregate data which is represented by a plurality of entities in the database.

This is clearly drawn out in each of the independent claims, by the inclusion of the features:

"providing an additional entity in the said database for said at least one set of entities"; and

Backup\patapps\614-L\614L_and2.doc

"storing in the additional entity of the aggregation of a plurality of data values contained in the said at least one set of linked entities whereby the aggregate data values are obtained by performing an aggregate operation on the additional entity".

That is, the claimed invention is directed to a system or method which preferably saves computational (database retrieval) time by avoiding the need to perform complex join operations when retrieving data from a database.

Krychniak, in contrast, while also seeking to make database access "more efficient", is not concerned with creating an additional entity and storing values in the additional entity, but rather, is concerned with a rewriting of a search query in order to perform a more efficient type of join operation.

Examiner contends that the feature of storing in aggregated data values in an additional entity "whereby the aggregated data values are obtained by performing a read operation on the additional entity" is disclosed at Column 2, Lines 11 to 28 of Krychniak.

Applicant contends that said Krychniak reference discloses no such feature. Column 2, Lines 11 to 28 of Krychniak refers to a particular and unusual situation, where it is recognised by a programmer that a join operation is not necessary.

Applicant provides extracts from Column 2, Lines 11 to 28 of Krychniak to further illustrate this point:

"If an initial mapping of the attribute value selected in each dimension is made on to the key values in that dimension, it is not necessary to join the dimension tables

with the fact table in the query, as all the necessary information is in the fact table".

In the abovementioned extract from the section of Krychniak quoted by the Examiner, the term "initial mapping" does not refer to the creation of a separate table arranged to contain aggregate data, but rather, refers to a particular implementation where the key values, which are normally arbitrary, are made equivalent to the attribute values, thereby not requiring a join operation in order to access the information in the fact table.

However, this paragraph is wholly silent on the creation of providing an "additional entity" which holds an aggregation of a plurality of data values.

Examiner cites Figure 1 as an example of the provision of an "additional entity". However, Examiner fails to point out where this additional entity resides in Figure 1. Figure 1 discloses a Fact Table (shown generally at the right hand side of the Figure), which is linked to Three Dimension Tables (seen generally at the left hand side of the Figure). There appears to be no additional entity (table) which contains an aggregation of the table, Fact Table and Three Dimensional Tables shown.

To reiterate, the fact that a join operation is not necessary in one particular instance (as recognised by a programmer), is not equivalent to providing an additional entity which includes aggregated data values that can be obtained by performing a read operation on the additional entity.

In other words, the fact that a join operation need not always be used in order to extract appropriate information from a database is not equivalent to

constructing an aggregate table within a database in a manner which ensures that no join operations need be performed in order to extract said aggregate data from the database.

Turning to the second substantive objection raised by the Examiner, Examiner contends that Claims 4, 5, 9 and 18 are obvious over *Krychniak* in view of *Prabhakaran* (US 6,859,758). (claim 18 has been cancelled and included into claim 9 as amended).

Examiner, in particular, contends that it would have been obvious to one of ordinary skill in the data processing art to combine the teachings of the cited references, because the teaching in *Prabhakaran* would have allowed *Krychniak* to measure the performance of the storage system. However, Examiner provides no reasoning as to why the skilled addressee would be likely to combine *Krychniak* with *Prabhakaran*.

Krychniak is wholly silent on methods and/or systems for determining read/write ratios. This is due to the fact that *Krychniak* is not concerned with calculating read/write ratios, but rather, is concerned with generating read queries which are more efficient (i.e. take less computing time to process).

In other words, *Krychniak* is only concerned with read operations. There is absolutely no teaching in *Krychniak* that read/write ratios are utilised or necessary in order to test the effectiveness or efficiency of the invention of *Krychniak*, as *Krychniak* does not seek to increase the performance of write queries, but rather is only concerned with read queries. Therefore, it is difficult to see how the disclosure of *Krychniak* could lead a skilled addressee to combine the two documents.

Notwithstanding the lack of motivation to combine, Applicant also submits that *Prabhakaran*, even when combined with

Krychniak, does not disclose all of the features of any one of Claims 4, 5, 9.

Claim 4, in particular, has been amended to more clearly define the claimed invention. Claim 4 now includes the further method steps of "comparing the initial read/write ratio of said database to a critical read/write ratio", where if said initial read/write ratio is greater than the critical read/write ratio, the method steps of Claim 1 are performed.

No such teaching is to be found in the *Prabhakaran* reference. There is no teaching, either implicit or explicit, in *Prabhakaran* of comparing a generated read/write ratio to a critical or desired ratio of reads to writes. Therefore, Claim 4 is novel and non-obvious, whether *Prabhakaran* is taken singularly or in combination with *Krychniak*.

Referring now to Claim 9, Examiner's attention is drawn wherein there is now seen the inclusion of the features of Claim 18 into amended Claim 9. *Prabhakaran* does not disclose the additional features of establishing a "critical read/write ratio".

Regarding the features of claim 18 (which are now part of claim 9), Examiner refers to Abstract of *Prabhakaran* as support for Examiner's contention that the features of Claim 18 are disclosed.

But note - the *Prabhakaran* "Abstract" makes no explicit reference to establishment of a "critical read/write ratio". Examiner refers to the phrase "preferably, the read and write commands adhere to a desired ratio of reads to writes".

This is not equivalent to a critical read/write ratio. In the disclosure of *Prabhakaran*, the phrase "desired ratio of reads to writes" refers to the read to write ratio used by the stress testing database software disclosed in *Prabhakaran*. It

is a "parameter" that is set by a user in order to stress test a database.

In Applicant's claimed invention, in contrast, the term "critical read/write ratio" refers to a value which is quantifiable and dependent on database performance.

In other words, the Prabhakaran phrase "desired ratio of reads to writes" is not equivalent to the term "critical read/write ratio" of Applicants.

As a result, it is respectfully requested that Examiner consider Applicant's claims as a whole in their entirety and subsequently provide a timely Notice of Allowance.

Respectfully submitted,

By Alfred W. Kozak
Alfred W. Kozak
Reg No. 24,265
(949) 380-5822

Dated: November 6, 2006

Unisys Corporation
25725 Jeronimo Road, MS400
Mission Viejo, CA 92691

CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8A)

I hereby certify that this correspondence is, on the date shown below, being:

MAILING

FACSIMILE

☐ deposited with the United States Postal Service as first class mail in an envelope addressed to:
Commissioner for Patents, PO Box 1450,
Alexandria, VA 22313-1450.

☒ transmitted by facsimile to the United States Patent and Trademark Office.

Date: November 6, 2006

Carol Wasserman November 6, 2006
Carol Wasserman Date